

- Justification for build action with Demeter God Card
 - What is a valid build action?
 - valid first build
 - valid second build
 - How to perform build?
 - first build
 - passing second build
 - second build
 - My design decisions
 - Alternatives considered and analysis of trade-offs

Justification for build action with Demeter God Card

What is a valid build action?

Suppose the current player has Demeter God Card, and the opponent has no special God Card.

valid first build

1. worker can only build on adjacent cells (including diagonals).
2. worker cannot build on a cell that is occupied by another worker, regardless of which player it belongs to.
3. worker cannot build on a cell that has a dome.

valid second build

1. worker can only build on adjacent cells (including diagonals).
2. worker cannot build on a cell that is occupied by another worker, regardless of which player
3. worker cannot build on a cell that has a dome.
4. worker cannot build on the same space as the first build.

How to perform build?

first build

First build is compulsory.

Game class stores buildable spaces for build turn phase. Only buildable spaces are playable and highlighted by the frontend for the player to choose from. The player select space and pass (x,y) parameter to Game's firstBuild method.

Game class calls Board's build method to get target space at (x,y). Board class delegates the build action to Space class, which modifies its own state (increasing tower level or adding dome) according to the build rules. Then Board returns a new Board instance with updated Space to Game class. Game class will compute the available build spaces for second build (excluding the space of first build) and finally return a new Game instance with updated Board, updated available build spaces and new turn phase.

passing second build

Second build is optional. Player can choose from frontend to skip second build. Game class will simply return a new Game instance with updated turn phase.

second build

Since updated available build spaces for second build have been computed before, the second build logic flow is almost the same as the first build. The only difference is that Game class will directly return a new instance with updated turn phase after getting updated Board.

My design decisions

1. **GodCard provides buildable spaces:** Game class call GodCard to provide buildable spaces based on the current state. Inside GodCard class, it will call Board to get spaces that are normally valid for building then deploy its own filter logic.

2. **Space handles building:** The build action modifies the state of a Space (increasing tower level or adding a dome). Therefore, it is logical for the Space class to actually update state.

Alternatives considered and analysis of trade-offs

1. **Assigning build responsibility to Player:** This would encapsulate all player actions within the Player class, but Player will take too much responsibility and every time Player need to access specific worker's position and state, which would bring unnecessary coupling.
2. **Assigning build responsibility to Board:** Since Board manages the spaces, and towers are not belonged to sepecific player or worker, it could handle building. However, this would mix managing the board state with player actions, reducing cohesion. It is more appropriate for Space to handle building.
3. **Getting buildable spaces directly in Game class:** Game class could handle god card logic directly to compute buildable spaces. However, this would violate single responsibility principle and lead to a Game class that is hard to maintain. Delegating to GodCard is much more sensible and adheres to open/closed principle.

